

Special Symbols in ex

.	Addressing symbol for current line
\$	Addressing symbol for last line of file
#	Alternate filename
%	Current filename
!	(after ex command) override warning

Alphabetical List of Keys

a	Append text after cursor.
A	Append text at end of line.
^A	Unused.
b	Back up to beginning of word in current line.
B	Back up to word, ignoring punctuation.
^B	Scroll backward one window.
c	Change operator.
C	Change to end of current line.
^C	Unused.
d	Delete operator.
D	Delete to end of current line.
^D	Scroll down half-window.
e	Move to end of word.
E	Move to end of word, ignoring punctuation.
^E	Show one more line at bottom of window.
f x	Find next x on current line.
F x	Find previous x on current line.
^F	Scroll Forward one window.
g	Unused.
G	Go to specified line or end of file.
^G	Print information about file on status line.
h	Left arrow cursor key.
H	Move cursor to Home position (top left of screen).
^H	Left arrow cursor key; backspace key in insert mode.
i	Insert text before cursor.
I	Insert text at beginning of line.
^I	Unused in command mode; in insert mode, same as TAB.
j	Down arrow cursor key.
J	Join two lines.
^J	Down arrow cursor key.
k	Up arrow cursor key.
K	Unused.
^K	Unused.
l	Right arrow cursor key.
L	Move cursor to Last position in window.
^L	Redraw screen (depends on terminal type).
m	Mark the current cursor position in register (a-z).
M	Move cursor to Middle position in window.
^M	Carriage return.
n	Repeat the last search command.
N	Repeat the last search command in reverse direction.
^N	Down arrow cursor key.

Alphabetical List of Keys (continued)

o	Open line below current line.
O	Open line above current line.
^O	Unused.
p	Put yanked or deleted text after or below cursor.
P	Put yanked or deleted text before or above cursor.
^P	Up arrow cursor key.
q	Unused.
Q	Quit vi and invoke ex.
^Q	Unused in command; in insert mode, quote next character.
r	Replace character at cursor with the next character typed.
R	Replace characters.
^R	Redraw screen (depends on terminal type).
s	Change the character under the cursor to typed characters.
S	Change entire line.
^S	Unused.
t	Find character before next x on current line.
T	Find character after previous x on current line character typed.
^T	Unused in command mode; in insert mode, used with autoindent option set.
u	Undo the last change made.
U	Restore current line, discarding changes.
^U	Scroll the screen upward half-window.
v	Unused.
V	Unused.
^V	Unused in command mode; in insert mode, quote next character.
w	Move to beginning of next word.
W	Move to beginning of next word, ignoring punctuation.
^W	Unused in command mode; in insert mode, back up to beginning of word.
x	Delete character at cursor.
X	Delete character before cursor.
^X	Unused.
y	Yank (or copy) operator.
Y	Make copy of current line.
^Y	Show one more line at top of window.
z	Redraw the screen, repositioning cursor when followed by RETURN at top, . at middle, and - at bottom of screen.
ZZ	Exit the editor, saving changes.
^Z	Unused.

All editing commands can take most movement commands as objects. These editing objects are indicated in bold. For instance, **dw** deletes a word **d)** deletes to the end of the sentence.

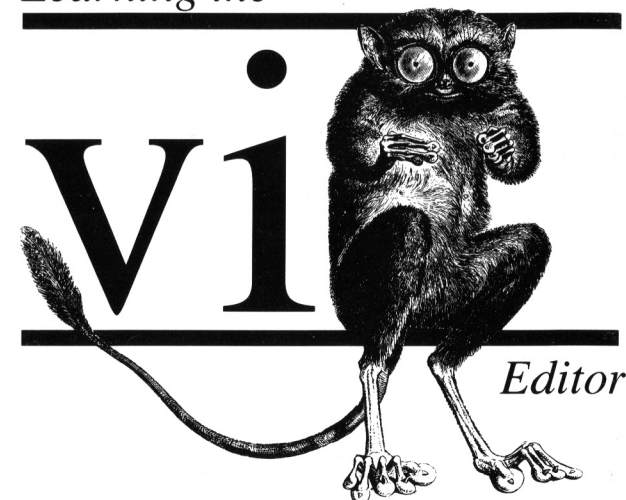
Most vi commands can be preceded by a count. For example, **5w** moves forward five words and **5dw** deletes to the next five words.

ex editing commands are entered by typing a colon, an optional address and the command. The address can be a line number, a pattern, or range of lines (*n,m*). With no address, the command is applied only to the current line.

© Copyright, 1990, O'Reilly & Associates, Inc. All rights reserved.

Quick Reference Guide

Learning the



O'Reilly & Associates, Inc.

Movement Commands

Character h, j, k, l	Left, down, up, right (or ←, ↓, ↑, →)
Text w, W, b, B e, E , ({, { }, { }, {	Forward, backward by word End of word Beginning of next, previous sentence Beginning of next, previous paragraph Beginning of next, previous section
Lines 0, \$ ^ +, - n H M L nH nL	First, last position of current line First character of current line (ignore spaces) First character of next, previous line Column <i>n</i> of current line Top line of screen Middle line of screen Last line of screen <i>n</i> (number) of lines after top line <i>n</i> (number) of lines before last line
Screens [CTRL-F], [CTRL-B] [CTRL-D], [CTRL-U] [CTRL-E], [CTRL-Y] Z [RETURN] Z Z-	Scroll forward, backward one screen Scroll down, up one-half screen Show one more line at bottom, top of window Reposition line with cursor: to top of screen Reposition line with cursor: to middle of screen Reposition line with cursor: to bottom of screen
Searches /pattern ?pattern n, N /, ? fx Fx tx Tx	Search forward for pattern Search backward for pattern Repeat last search in same, opposite direction Repeat previous search forward, backward Search forward for character <i>x</i> in current line Search backward for character <i>x</i> in current line Search forward for character before <i>x</i> in current line Search backward for character after <i>x</i> in current line
; (semicolon) , (comma)	Repeat previous current-line search Repeat previous current-line search in opposite direction
Line number [CTRL-G] nG G :n	Display current line number Move to line number <i>n</i> Move to last line in file Move to line number <i>n</i>
Marking position mx `x `` 'x ''	Mark current position as <i>x</i> Move cursor to <i>x</i> Return to previous mark or context Move to beginning of line containing mark <i>x</i> Return to beginning of line containing previous mark

Editing Commands

Insert i, a I, A o, O	Insert text before, after cursor Insert text at beginning, end of line Open new line for text below, above cursor
Change* r ~ Cm cc C R S S	Replace with next typed character Change case of character Change text block defined by movement command <i>m</i> (e.g., cW or cL) Change current line Change to end of line Type over characters Delete character and substitute text Delete current line and substitute text
* Note: r and ~ leave you in command mode	
Delete, move x X dm dd D p, P "np	Delete character Delete character before cursor Delete text block defined by movement command <i>m</i> (e.g., d/pattern or dG) Delete current line Delete to end of line Put deleted text after, before cursor Put text from delete buffer number <i>n</i> after cursor (for last nine deletions)
Yank ym yy, Y "ayy p, P "aP	Yank (copy) text block defined by movement command <i>m</i> (e.g., y]] or y}) Yank current line Yank current line into named buffer <i>a</i> Put yanked text after, before cursor Put text from buffer <i>a</i> before cursor
Other commands . u, U J [CTRL-L], [CTRL-R]	Repeat last edit command Undo last edit; restore current line Join two lines Redraw screen
ex equivalents :d :m :co or :t :.,\$d :30,60m. :., /pattern /co\$	Delete lines Move lines Copy lines Delete from current line to end of file Move lines 30 through 60 after current line Copy from current line through line containing pattern to end of file
Command Line Options	
vi file vi file1 file2 view file	Invoke vi editor on <i>file</i> Invoke vi editor on files sequentially Invoke vi editor on <i>file</i> in read-only mode

Command Line Options (continued)

vi -R file vi -r file vi + file vi +n file vi +/pattern file ex file	Invoke vi editor on <i>file</i> in read-only mode Recover <i>file</i> and recent edits after system crash Open <i>file</i> at last line Open <i>file</i> directly at line number <i>n</i> Open <i>file</i> directly at <i>pattern</i> Invoke ex editor on <i>file</i> from UNIX
Other ex Commands	
ex Commands : :sh ^D :! command :r newfile :r ! command	Invoke ex from vi editor Invoke shell Return to editor from shell Give UNIX <i>command</i> Read contents of <i>newfile</i> into current file Read output of UNIX <i>command</i> into current file
Abbreviations* :map x sequence :unmap x :map! x sequence :unmap! x :ab abbr phrase :unab abbr	Define keystroke <i>x</i> as a command sequence Disable the map <i>x</i> Define <i>x</i> as command <i>sequence</i> invoked in insert mode Disable insert mode map <i>x</i> Abbreviate <i>phrase</i> as <i>abbr</i> ; when <i>abbr</i> is typed in insert mode, it expands to full words or phrases Disable abbreviation
Setting Options* :set option :set option=value :set nooption :set :set all :set option?	Activate <i>option</i> Assign <i>value</i> to <i>option</i> Deactivate <i>option</i> Display options set by user Display list of all current options, both default and those set by the user Display value of <i>option</i>
* Note: in .exrc file, omit colon for ex command.	
Exit Commands	
ZZ :x :wq :w :w! :30,60w newfile :30,60w>> file :w %. new :q :q! Q :e file2 :n :e! :e#	Write (save) and quit file Write (save) and quit file Write (save) and quit file Write (save) file Write (save) file (overriding protection) Write from line 30 through line 60 as <i>newfile</i> Write from line 30 through line 60 and append to <i>file</i> Write current buffer named <i>file</i> as <i>file.new</i> Quit file Quit file (discarding any changes) Quit vi and invoke ex Edit <i>file2</i> without leaving vi Edit next file Return to version of current file at time of last write (save) Edit alternate file